

Popover

The <ava-popover> component provides a flexible overlay system for displaying contextual information, help text, and guided tour content. Built with advanced positioning logic and multiple navigation variants, it's perfect for creating interactive user experiences.

How to use

Basic Usage

Simple popover implementation with default settings and basic content display.

```
import { AavaPopOverComponent } from "@aava/play-core";
```

Basic Features

- Flexible Positioning : Automatic positioning with arrow alignment
- Content Management : Support for header, description, and optional learn more links
- Responsive Design : Adapts to different screen sizes and content lengths
- Accessibility : Built-in accessibility features and keyboard navigation
- Animation : Smooth entrance animations with stretch effects

Positioning Options

Four positioning variants with intelligent arrow placement and automatic boundary detection.

```

<button aava-button
  avaPopover
    [avaPopoverData]="samplePopoverData"
    [avaPopoverPosition]='top'
    [avaPopoverArrow]='center'
    [label]='Show Popover'
    [variant]='primary'
    [outlined]="true"
    pressedEffect="ripple"
</button>
<button aava-button
  avaPopover
    [avaPopoverData]="learnMorePopoverData"
    [avaPopoverPosition]='bottom'
    [avaPopoverArrow]='center'
    [avaPopoverShowLearnMore]="true"
    [label]='Show with Learn More'
    [variant]='secondary'
    [outlined]="true"
    pressedEffect="ripple"
</button>
<button aava-button
  avaPopover
    [avaPopoverData]="multiStepPopoverData"
    [avaPopoverPosition]='right'
    [avaPopoverArrow]='center'
    [avaPopoverShowButtons]="true"
    [avaPopoverShowPagination]="true"
    [label]='Show Multi-step'
    [variant]='success'
    [outlined]="true"
    pressedEffect="ripple"
</button>

```

Available Positions

- Top - Appears above the target element with downward arrow
- Bottom - Appears below the target element with upward arrow
- Left - Appears to the left of the target element with rightward arrow
- Right - Appears to the right of the target element with leftward arrow

Arrow Alignment

- Start - Arrow positioned at the beginning of the popover
- Center - Arrow centered on the popover (default)
- End - Arrow positioned at the end of the popover

API Reference

Inputs

Property	Type	Default	Description
config	PopoverConfig	-	Configuration object for popover behavior
data	PopoverData[]	[]	Array of content items to display

PopoverConfig Interface

Property	Type	Default	Description
arrow	'start' 'center' 'end' null	null	Arrow alignment position
position	'top' 'bottom' 'left' 'right'	'top'	Popover position relative to target
left	number	0	Target element left position
top	number	0	Target element top position
bottom	number	0	Target element bottom position
width	number	0	Target element width
height	number	0	Target element height
showButtons	boolean?	false	Whether to show navigation buttons
showPagination	boolean?	false	Whether to show page counter
showIcon	boolean?	false	Whether to show icon navigation

Property	Type	Default	Description
showSkip	boolean?	false	Whether to show skip button
showLearnMore	boolean?	false	Whether to show learn more button

PopoverData Interface

Property	Type	Default	Description
header	string	-	Header text for the popover
description	string	-	Description text for the popover
learnMoreUrl	string?	-	Optional URL for learn more functionality

Properties

Property	Type	Description
visible	boolean	Whether the popover is currently visible
currentIndex	number	Current page index for multi-page content
currentData	PopoverData null	Current content data being displayed
currentPage	string	Current page display string (e.g., "2/5")
hasPrevious	boolean	Whether previous page is available
hasNext	boolean	Whether next page is available

Methods

Method	Parameters	Return Type	Description
previous()	event?: any	void	Navigate to previous page
next()	event?: any	void	Navigate to next page
skip()	event?: any	void	Close popover and skip content
learnMore()	event?: any	void	Open learn more URL in new tab
updatePosition()	-	void	Update popover position and arrow placement
setArrowPosition()	-	void	Set CSS custom properties for arrow positioning

CSS Custom Properties

The component uses CSS custom properties for theming and styling:

Property	Default	Description
--popover-background	Dynamic	Background color of the popover
--popover-text	Dynamic	Text color of the popover
--popover-font	Dynamic	Font family for the popover
--popover-padding	Dynamic	Internal padding of the popover
--popover-border-radius	Dynamic	Border radius of the popover

Property	Default	Description
--popover-max-width	Dynamic	Maximum width of the popover
--popover-z-index	Dynamic	Z-index for layering
--popover-shadow	Dynamic	Box shadow for the popover
--popover-border-color	Dynamic	Border color of the popover
--popover-content-gap	Dynamic	Gap between content sections
--popover-header-font-family	Dynamic	Font family for the header
--popover-header-font-size	Dynamic	Font size for the header
--popover-header-font-weight	Dynamic	Font weight for the header
--popover-header-color	Dynamic	Color for the header text
--popover-header-line-height	Dynamic	Line height for the header
--popover-description-font-family	Dynamic	Font family for the description
--popover-description-font-size	Dynamic	Font size for the description
--popover-description-font-weight	Dynamic	Font weight for the description
--popover-description-color	Dynamic	Color for the description text
--popover-description-line-height	Dynamic	Line height for the description
--popover-footer-margin-top	Dynamic	Top margin for the footer
--popover-pagination-font-size	Dynamic	Font size for pagination text

Property	Default	Description
--popover-pagination-font-weight	Dynamic	Font weight for pagination text
--popover-pagination-color	Dynamic	Color for pagination text
--popover-pagination-font-family	Dynamic	Font family for pagination text
--popover-pagination-line-height	Dynamic	Line height for pagination text
--popover-button-gap	Dynamic	Gap between navigation buttons
--popover-button-width	Dynamic	Width of navigation buttons
--popover-button-height	Dynamic	Height of navigation buttons
--popover-disabled-opacity	Dynamic	Opacity for disabled button states
--popover-hover-background	Dynamic	Background color for button hover states
--popover-hover-border-radius	Dynamic	Border radius for button hover states
--popover-arrow-border-width	Dynamic	Width of arrow border
--popover-arrow-border-width-inner	Dynamic	Width of inner arrow border
--popover-arrow-border-shadow	Dynamic	Color of arrow border shadow
--popover-arrow-color	Dynamic	Color of arrow fill

Best Practices

Content Guidelines

- Keep it concise : Popovers work best with short, focused content

- Use clear headers : Make the purpose of the popover immediately clear
- Provide actionable content : Include learn more links or navigation when appropriate
- Consider context : Position popovers to avoid covering important interface elements

Positioning Guidelines

- Choose appropriate positions : Use top/bottom for horizontal content, left/right for vertical content
- Align arrows thoughtfully : Center arrows for balanced content, start/end for directional emphasis
- Avoid boundaries : Ensure popovers don't extend beyond viewport edges
- Consider scrolling : Position popovers to remain visible during page scroll

Navigation Guidelines

- Match user expectations : Use familiar navigation patterns (prev/next, pagination)
- Provide clear feedback : Show current position and available actions
- Include skip options : Allow users to bypass content when appropriate
- Use consistent styling : Maintain visual consistency across navigation elements

Technical Notes

Component Architecture

The popover component uses a modern Angular architecture with:

- PopOverComponent - Main component with positioning and navigation logic
- IconComponent integration for icon-based navigation
- ViewChild references for DOM manipulation and positioning
- ChangeDetectionStrategy.OnPush for performance optimization

Positioning System

The component implements an advanced positioning system:

- Automatic positioning based on target element coordinates
- Arrow alignment with CSS custom properties for precise placement
- Boundary detection to prevent overflow beyond viewport
- Scroll and resize handling for responsive positioning

Animation System

The component provides smooth entrance animations:

- Stretch effects for natural expansion from target elements
- Position-based animations with appropriate transform origins
- CSS keyframes for consistent and performant animations
- Smooth transitions for opacity and transform changes

Event Handling

The component manages multiple event types:

- Navigation events for page changes and content progression
- User interaction for skip, learn more, and navigation actions
- Window events for scroll and resize handling
- Lifecycle events for proper cleanup and memory management

CSS Architecture

The component uses a comprehensive CSS architecture:

- CSS custom properties for theming and customization
- BEM-like naming for maintainable and scalable styles
- Position-based modifiers for different placement variants
- Responsive design with flexible layouts and sizing

Accessibility Features

The component includes built-in accessibility features:

- Keyboard navigation for all interactive elements
- ARIA attributes for screen reader compatibility
- Focus management for proper tab order and focus indicators
- Semantic markup for clear content structure and meaning